

Appln No. 10/004,246  
Amdt date May 2, 2005  
Reply to Office action of April 8, 2005

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1.-14. (Cancelled)

15. (Currently Amended) The method of claim ~~14~~ 16, wherein executing the first instruction loads a plurality of configurations into respective configuration registers, wherein one of the plurality of configurations is loaded into a configuration register.

16. (Previously Presented) A method of controlling a reconfigurable processor, comprising:

executing a first instruction that loads a configuration into a configuration register;

executing a second instruction that references the configuration register; and

executing the configuration in the configuration register referenced by the second instruction, wherein executing the first instruction loads a plurality of configurations into respective configuration registers, wherein one of the plurality of configurations is loaded into a configuration register, and wherein the configuration and the first instruction are stored in a memory, and wherein the first instruction includes a displacement field indicating a location in the memory of the configuration relative to the first instruction.

Appln No. 10/004,246  
Amdt date May 2, 2005  
Reply to Office action of April 8, 2005

17. (Currently Amended) The method of claim~~-14~~ 16, wherein an application program issues the first instruction.

18. (Currently Amended) The method of claim~~-14~~ 16, wherein a compiler generates the first instruction.

19. (Currently Amended) The method of claim~~-14~~ 16, wherein executing the second instruction and the configuration further comprises retrieving operands requested by the second instruction and the configuration.

20. (Original) The method of claim 19, wherein the second instruction provides the operands to the configuration.

21. (Original) The method of claim 19, wherein a register provides the operands to the configuration.

22. (Original) The method of claim 19, wherein the second instruction includes an immediate value field, the second instruction being executed with values stored in the immediate value field.

23. (Original) The method of claim 19, wherein the second instruction includes an immediate value field, the configuration being executed with values stored in the immediate value field.

**Appln No. 10/004,246**

**Amdt date May 2, 2005**

**Reply to Office action of April 8, 2005**

24. (Currently Amended) The method of claim ~~14~~ 16, further comprising:

decoding controls from the second instruction and the configuration; and

processing data according to the decoded controls with one or more execution units in parallel.

25. (Original) The method of claim 24, further comprising generating one or more results with the one or more execution units.

26. (Original) The method of claim 25, further comprising writing the one or more results to a register.

27. (Original) The method of claim 25, further comprising storing the one or more results to a memory.

28. (Original) The method of claim 25, further comprising providing the one or more results to respective execution units.

29. (Currently Amended) The method of claim ~~14~~ 16, further comprising pre-loading a second configuration register with a configuration while the configuration previously loaded in the first configuration register executes.

30. (Currently Amended) The method of claim ~~14~~ 16, further comprising stalling the second instruction while the referenced configuration register is being loaded with a configuration.

Appln No. 10/004,246

Amdt date May 2, 2005

Reply to Office action of April 8, 2005

31. (Currently Amended) The method of claim~~14~~ 16, wherein the first instruction, the second instruction, and the configuration are executed as part of an application program.

32. (Currently Amended) The method of claim~~14~~ 16, wherein executing the second instruction and the configuration includes performing an operation on scalar data.

33. (Currently Amended) The method of claim~~14~~ 16, wherein executing the second instruction and the configuration includes performing an operation on vector data.

34. (Currently Amended) The method of claim~~14~~ 16, wherein executing the second instruction and the configuration includes performing an operation on scalar data and performing an operation on vector data.

35. (Canceled)

36. (Currently Amended) A processing system, comprising:  
means for executing a first instruction that loads a configuration into a configuration register; and  
means for decoding a second instruction and the configuration, the second instruction referencing the configuration register containing the configuration  
means for executing the second instruction and the configuration in parallel, wherein one of the plurality of

Appln No. 10/004,246

Amdt date May 2, 2005

Reply to Office action of April 8, 2005

configurations is loaded into a configuration register, and wherein the configuration and the first instruction are stored in a memory, and wherein the first instruction includes a displacement field indicating a location in the memory of the configuration relative to the first instruction.

37-83. (Canceled)

84. (Original) A method of implementing a vector processing system, comprising:

executing a first instruction that loads a configuration into a configuration register;

executing a second instruction and a configuration stored in a configuration register referenced by the second instruction;

processing elements of a first vector according to the second instruction and the configuration, wherein

a vector register stores elements of the first vector, and

a vector address unit provides an address to the vector register which stores the first vector elements selected by the second instruction and the configuration.

85. (Original) The method of claim 84, wherein processing elements of the first vector further comprises writing data to the identified address through a write port of the vector register file.

**Appln No. 10/004,246**

**Amdt date May 2, 2005**

**Reply to Office action of April 8, 2005**

86. (Original) The method of claim 84, wherein processing elements of the first vector further comprises reading data from the identified address through a read port of the vector register file.

87. (Original) The method of claim 84, wherein processing elements of the first vector further comprises:

initializing a current address of the first vector with a start address;

processing a first element of the first vector referenced by the current address with the instruction and configuration.

88. (Original) The method of claim 87, further comprising:

incrementing the current address with an address stride, wherein the incremented current address represents an address of a second element of the first vector; and

processing the second element referenced by the incremented current address.

89. (Original) The method of claim 88, for each successive element of the first vector, further comprising:

incrementing the previous current address with the address stride resulting in a new current address, wherein each successive new current address represents an address of a successive vector element; and

processing each successive vector element until all of the elements of the first vector have been processed.

**Appln No. 10/004,246**

**Amdt date May 2, 2005**

**Reply to Office action of April 8, 2005**

90. (Original) The method of claim 89, further comprising identifying a start address of a second vector.

91. (Original) The method of claim 90, wherein identifying the start address of the second vector further comprises incrementing the start address of the first vector with a frame stride resulting in a second start address, wherein an initial value of a current address comprises the second start address.

92. (Original) The method of claim 91, further comprising processing the vector element referenced by the current address of the second vector.

93. (Original) The method of claim 92, for each successive vector element of the second vector, further comprising:

incrementing the previous current address with the address stride resulting in a new current address, wherein each successive new current address represents an address of a successive vector element of the second vector; and

processing each successive vector element until all of the elements of the second vector have been processed.

94. (Original) The method of claim 93, for each vector to be processed, further comprising:

identifying a start address of the vector;

processing a first element of the vector;

processing remaining successive elements of the vector by

**Appln No. 10/004,246**

**Amdt date May 2, 2005**

**Reply to Office action of April 8, 2005**

incrementing the current address with an address stride resulting in successive current addresses;

processing corresponding successive elements referenced by the successive current addresses; and

after all of the elements of the vector have been processed, incrementing the start address by the frame stride to identify a start address of the next vector to be processed.

95. (Original) The method of claim 84, wherein a vector of data elements is loaded into the vector file prior to execution of the second instruction and the configuration.

96. (Original) The method of claim 84, wherein a vector of data elements is loaded into the vector file in parallel with execution of the second instruction and the configuration.

97. (Original) The method of claim 96, wherein the first instruction operates to process the first vector element by element by interlocking between the load port of the vector register file and the vector computation to process each element when it arrives in the vector register file.